Parveen Khurana

124 Followers About Following

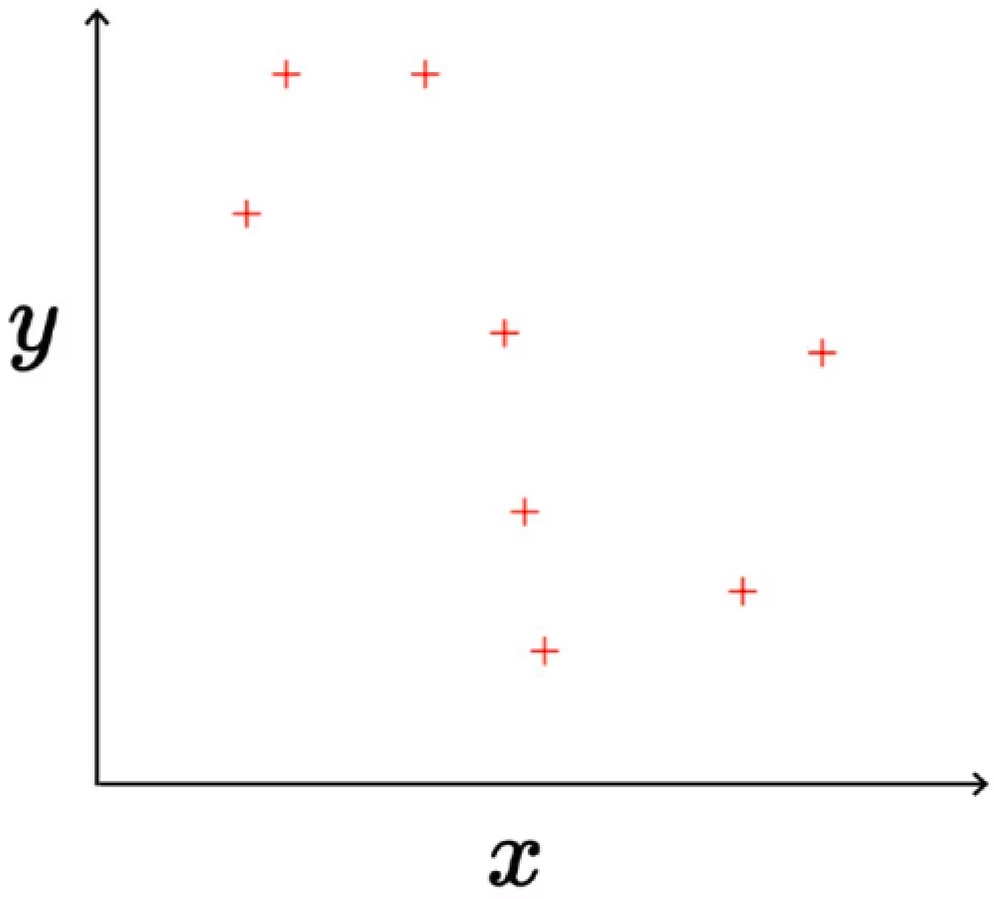
Bias — Variance Tradeoff

P Parveen Khurana Feb 5, 2020 · 11 min read

This article covers the content discussed in the Regularization Methods module of the Deep Learning course and all the images are taken from the same module.

Bias — Variance Trade-Off:

Let's say we have some toy data and below is the representation of the same:



toy data from ourselves, let's say we know the true relationship that exists between **'x'** and **'y'**.

True Relation*

$$y = f(x)$$

***In this case I know that $f(x) = sin(x)$**

Our job in ML/DL is to approximate the relation between **'x'** and **'y'** using a function. So, we are going to make two approximations: one is the simple one where we approximate the relation between **'x'** and **'y'** as a linear function and the other is the complex one(based on the number of parameters as well as because of the form of the function) where we approximate the relation between **'x'** and **'y'** as a polynomial of degree 25.

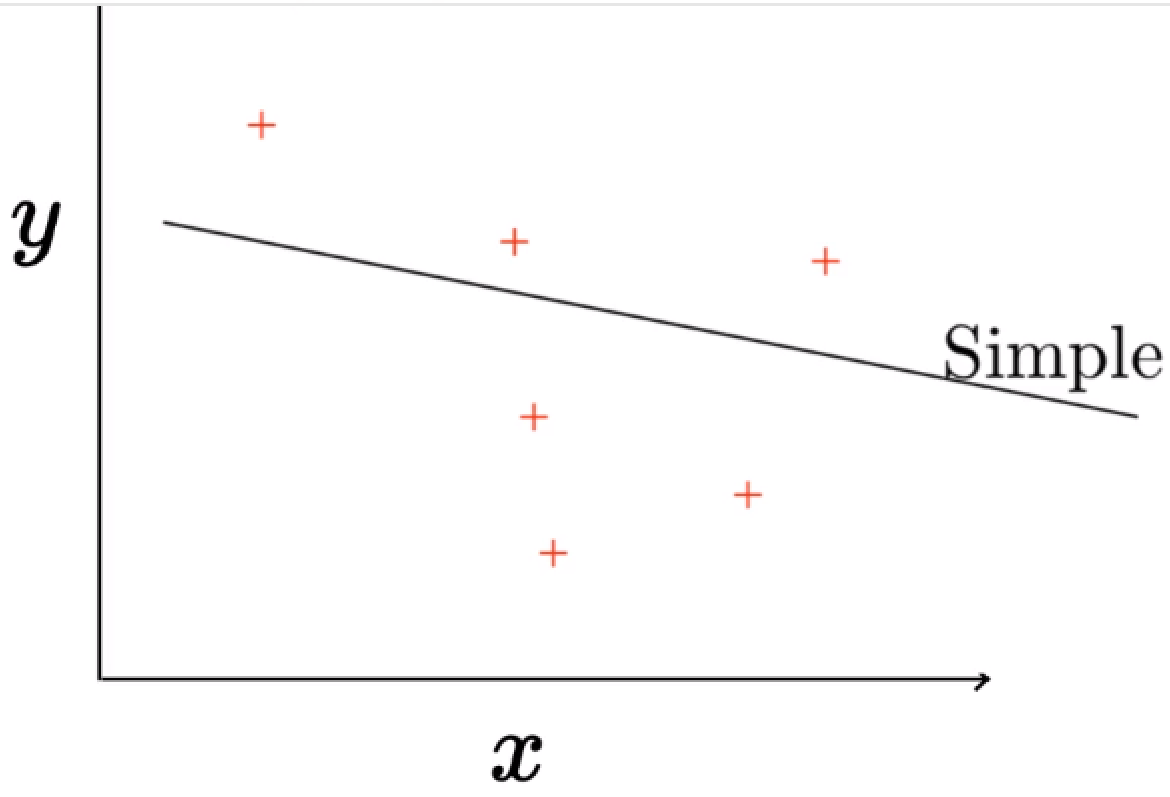
Our Approximation(model):

Simple (degree:1) $$y = \hat{f}(x) = w_1 x + w_0$$

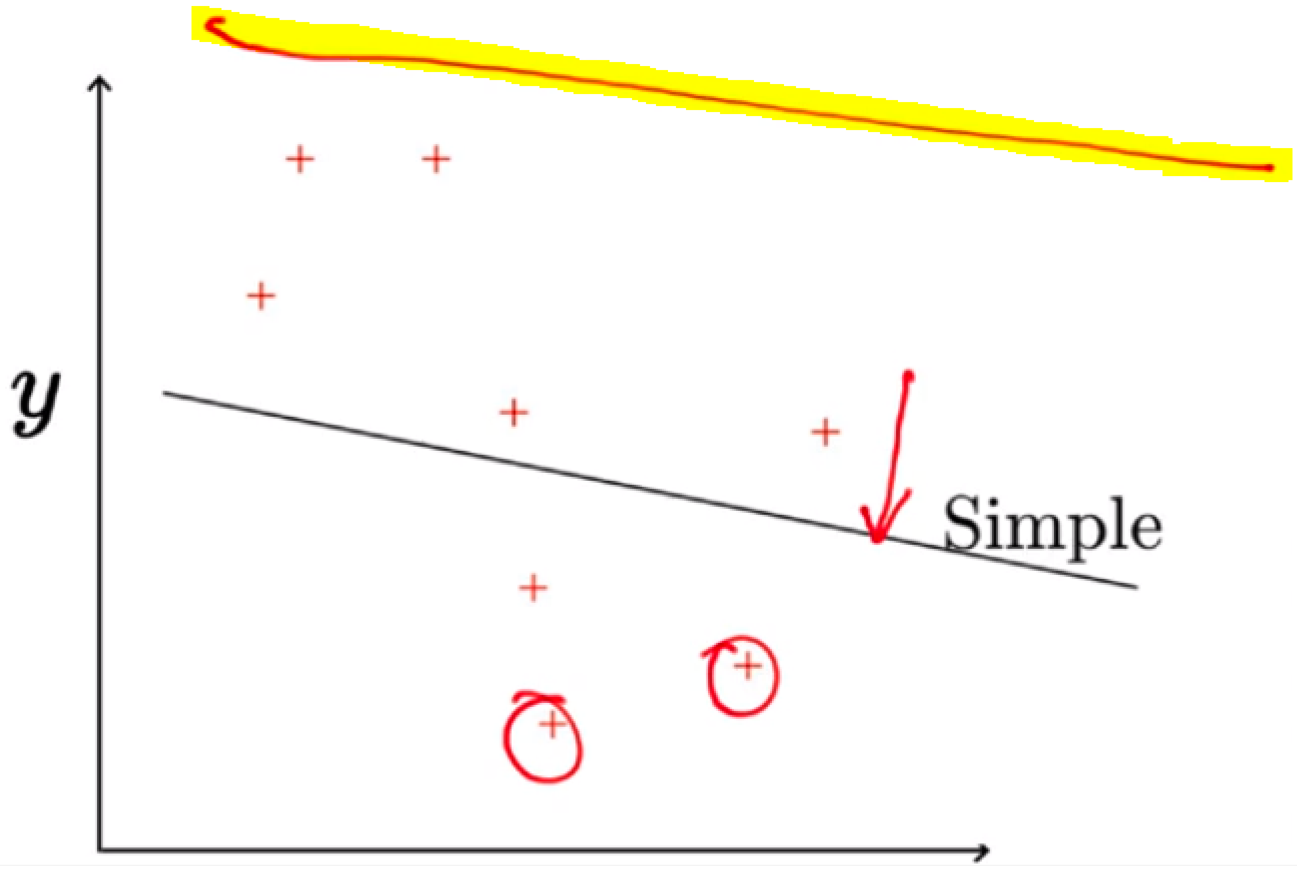
Complex (degree:25) $$y = \hat{f}(x) = \sum_{i=1}^{25} w_i x^i + w_0$$

So, using both the models, we compute the predicted output, compute the loss using say squared error loss function and then based on that we will learn the parameters using the Gradient Descent Algorithm in such a way that the loss value is minimized.

In the Simple case, the model would look like:

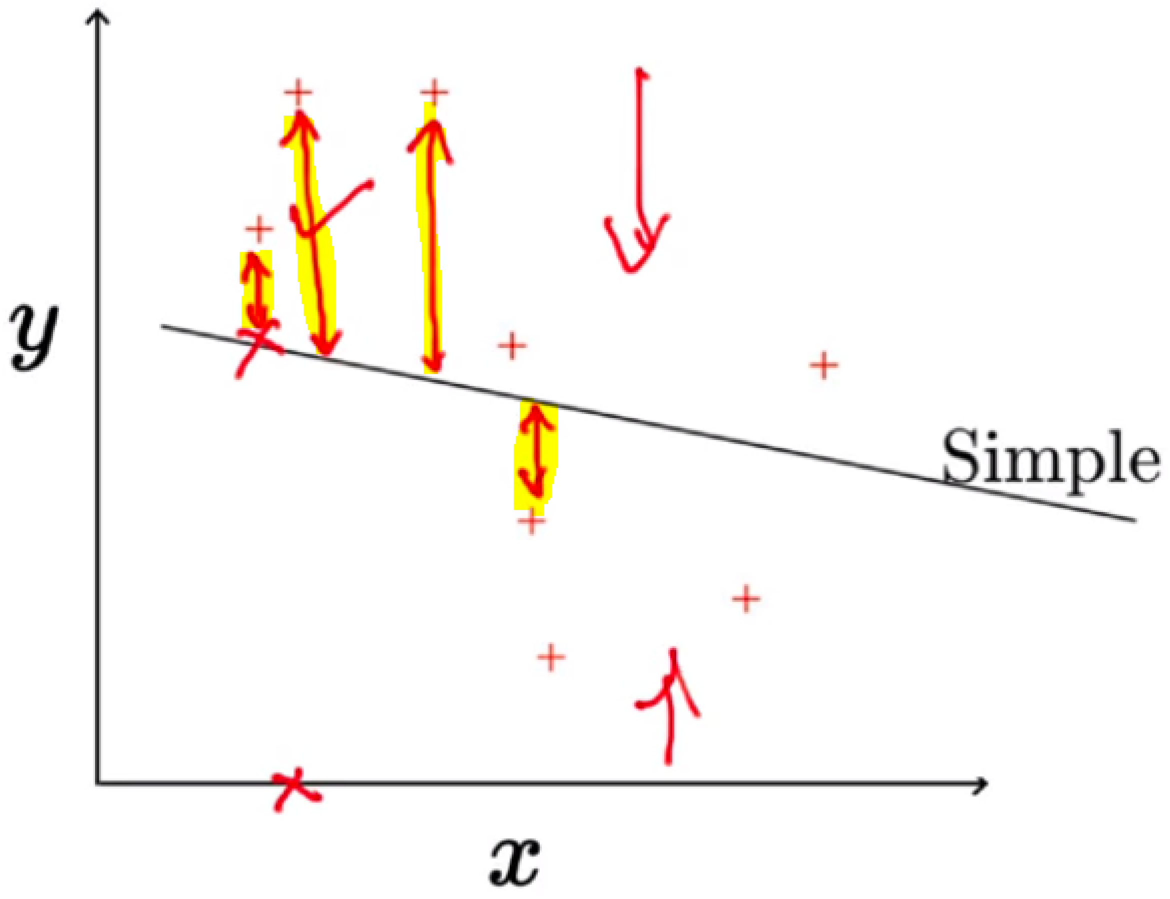


And it's obvious what the model has learned here, the only way it's going to minimize the error is to have a line so that on average it's closer to all points. If we draw the line in a way as the yellow highlighted one in the below image:

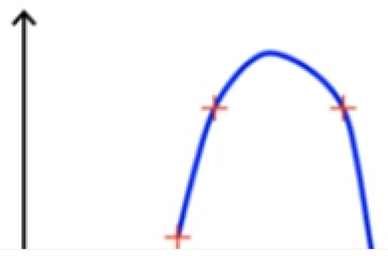


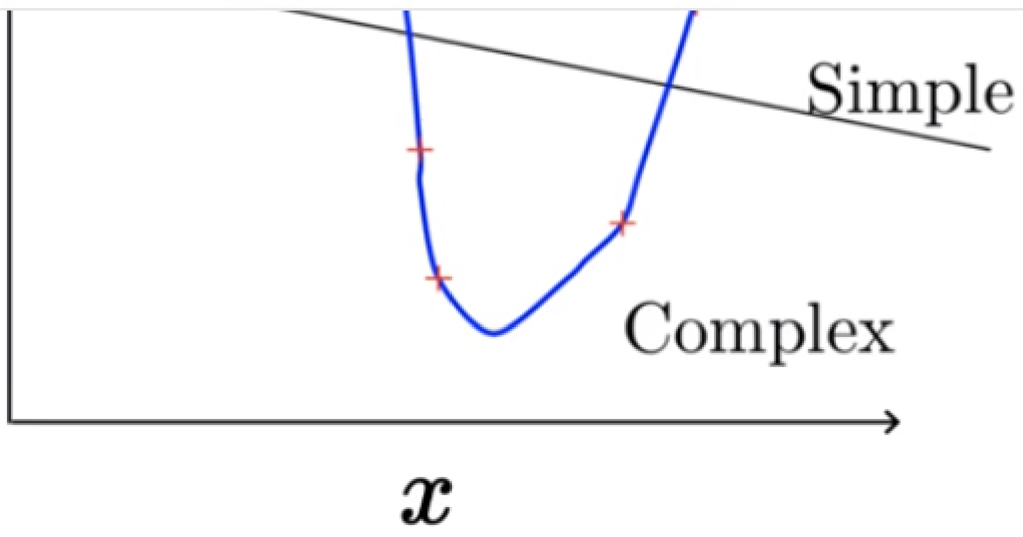
then it is going to be far away from the red circled points. So, what makes sense and what the model has also learned is to draw a line which roughly passes through the center of the data, there are some points on the upper half and there are some points on the lower half and that's how it has learned this line.

The overall error the model is making would be the squared sum of the difference(shown in yellow for some points in the below image) between the true output and the predicted output.



For the complex model, the plot would look like:



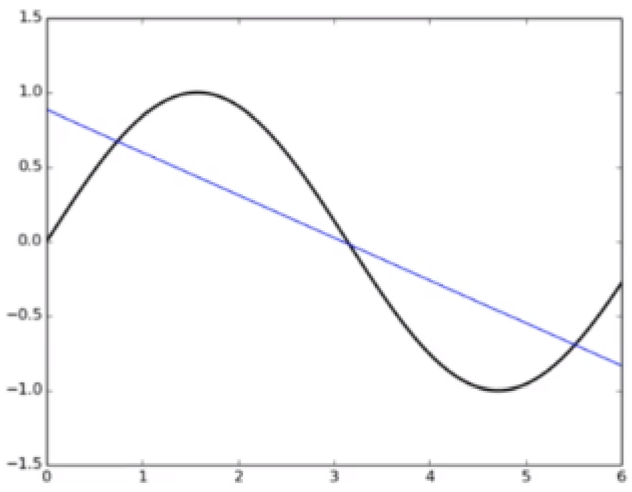


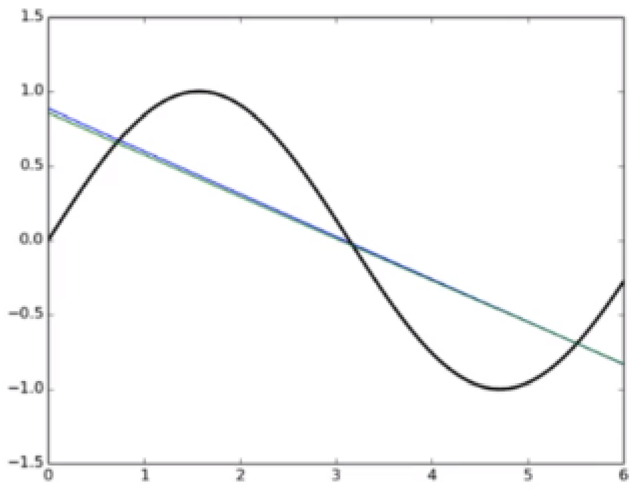
Here for the complex model also, the objective was to fit the training data in a way that the overall loss is minimized and the model has done a good job in learning the data and is doing good on the training data.

Let’s see what happens if we train both the models(both simple one and complex one) on different subsets of training data.

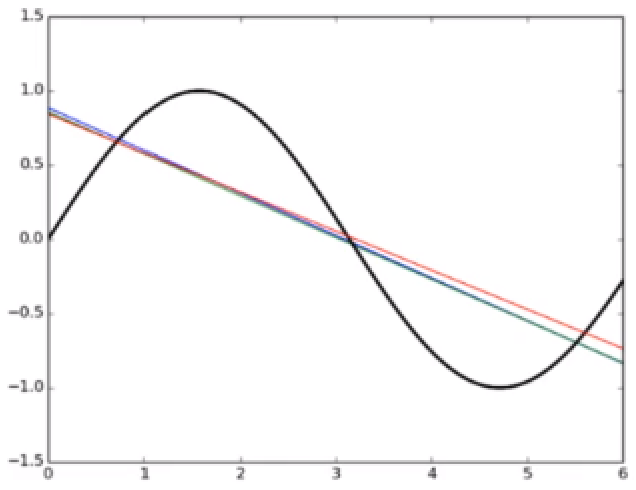
Consider we have 1000 data points all of which are shuffled and we take 100 data points randomly and train the model for these 100 points and we repeat this a few times.

So, for a simple function as we know our model is linear, the plot would look like(line in the below plot, the curve is the actual/true relationship that exists between the input and the output):



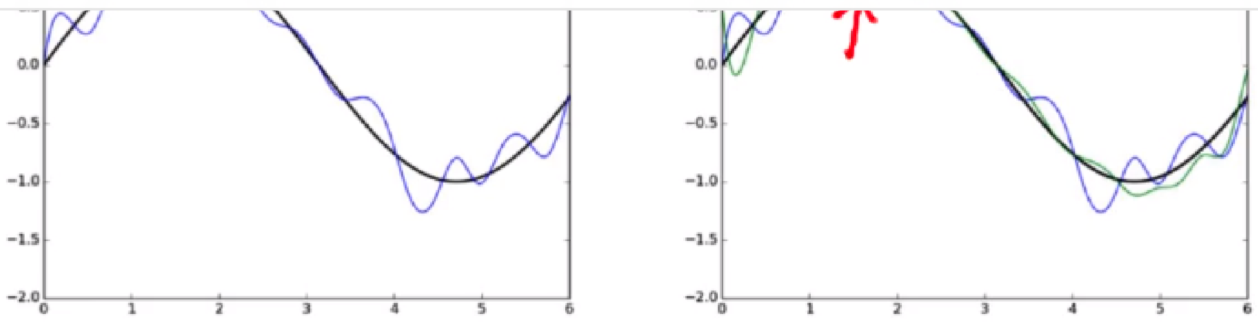


So, we can see that the line that we got, in this case, is very close to the previous line. **That means if we learn the parameters of a simple model using different subsets of training data, there is not much difference in the parameters the model learned in both the cases.** And if we try it again to learn the parameters using a different set of 100 randomly chosen points, we again get the line that is close to the other lines.



In other words, a simple model is not very sensitive to the training data given to it. It's trying to learn a similar line as in the above case.

If we do the same experiment with a complex model, we get the below results:

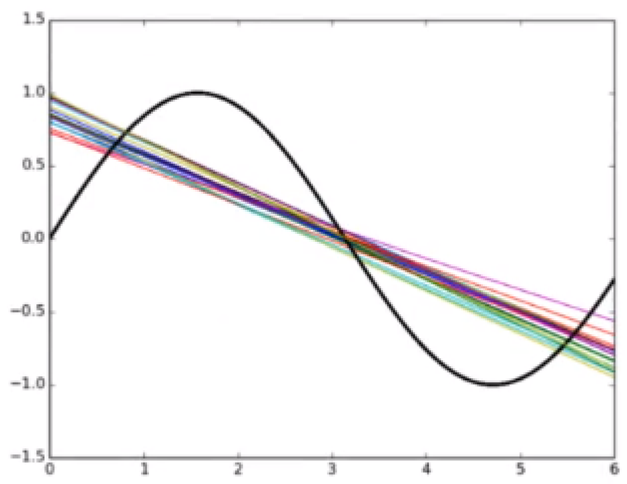


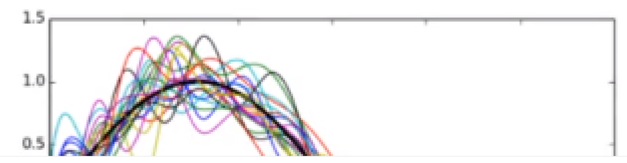
We can see that although the family of functions is the same the parameter values are different for each case where we are taking 100 points randomly and are training the model. So, in each of the cases, depending upon the 100 points we are given, the model tries to come up with a curve that best explains these 100 points that we have given it. And as we give it different-different 100 points, it will give us a curve which is going to look very very different from the other curves because each one will try to fit the 100 points it has.

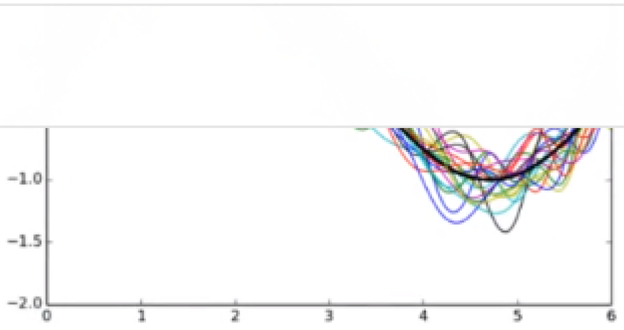
So, a complex model gives different values of parameters for different subsets of the training data whereas a simple model would give back almost the same value of parameters for different subsets of training data.

Bias and Variance:

The below image shows the plots of all the simple or complex models that we got on the subsets of the training data:







For the simple model, all the lines are very very close to each other forming a narrow band whereas, in the case of a complex model, all the polynomials are very very different from each other.

All the lines in case of a simple model are very close to each other but all of them are very far from the true curve that we have whereas, in the case of complex model, all the curves are very different from each other whereas all of them on average is still very close to the true curve reflecting the true relationship between the input and the output.

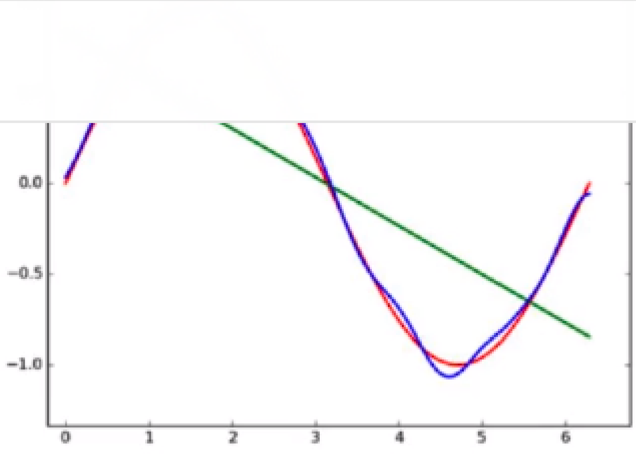
Bias is the difference between the Expected value of the model and the true value.

$$\text{Bias}\ (\hat{f}(x)) = E[\hat{f}(x)] - f(x)$$

Let's take the value of **'x'** as 90 degrees, the true output would be 1 as the true function is **sine** function which would give value as 1 for input as 90 degrees. Now **we have tried say 25 models using different subsets of the training data, the average of the output given by each of these 25 models for the same input say 90 degrees, in this case, is termed as the Expected value of the model.**

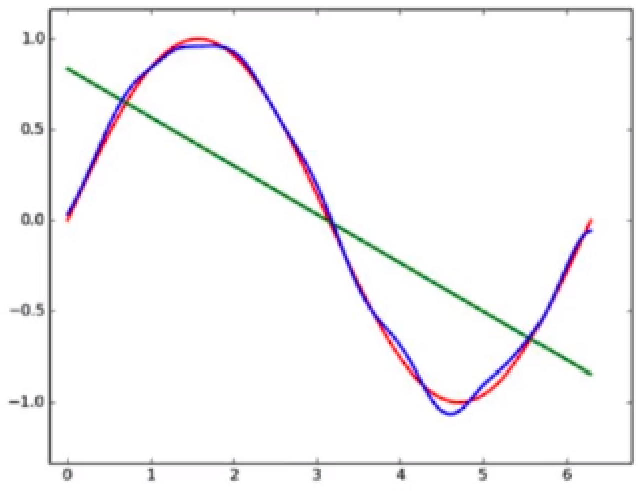
So, we are trying to see how far are our predictions from the true predictions no matter how many models we try.

The green line in the below image is the average of all the simple models(average of all the lines) that we have tried:



And what it reflects is that the green line is actually very far from the true predictions and hence the bias of this model is going to be high. On average, the average value is going to be far off from the true value.

If we do the same thing on a complex model and plot it's average it would look like the blue curve(blue curve is the average of say 25 curves/models that we got after training our complex model on different subsets of training data) in the below image and it is clear that the average value or expected value is close to the true value.



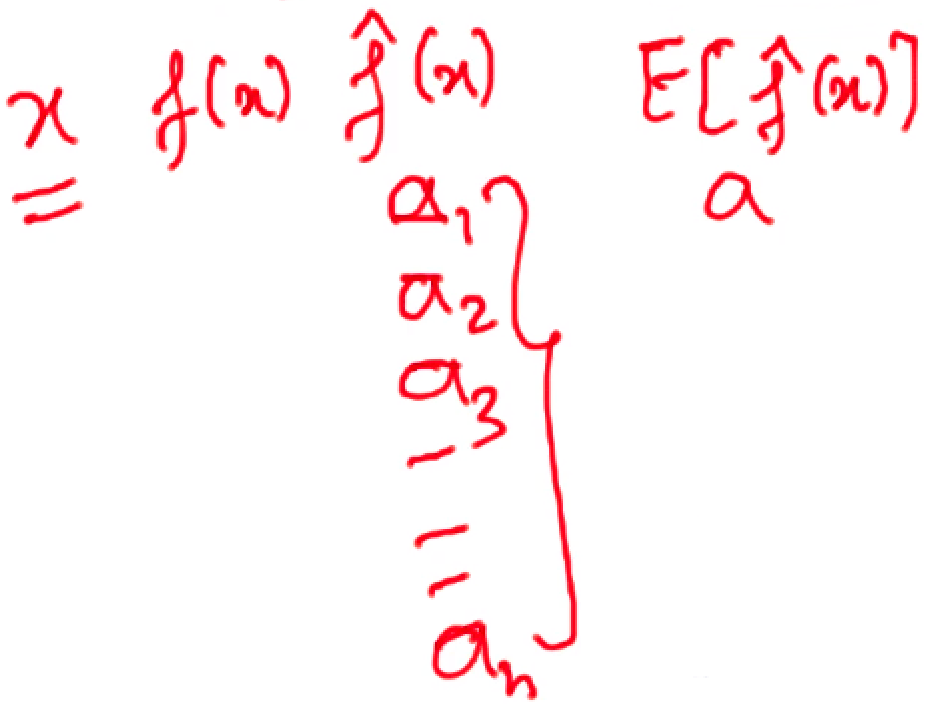
So, **it is very clear at this point that a simple model has high bias and the complex model has low bias.**

Variance:

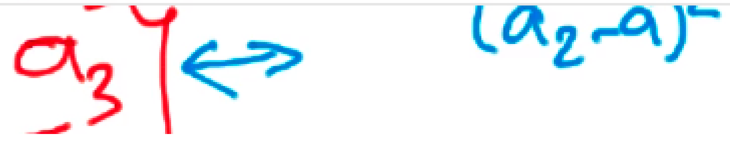
$$\text{Variance }(\hat{f}(x)) = E[(\hat{f}(x) - E[\hat{f}(x)])^2]$$

inside the bracket(highlighted in the above image). It is just the average values across all the different model plots that we get by training over different subsets of training data.

The second expectation is actually an expectation over all the training points that we are going to take.

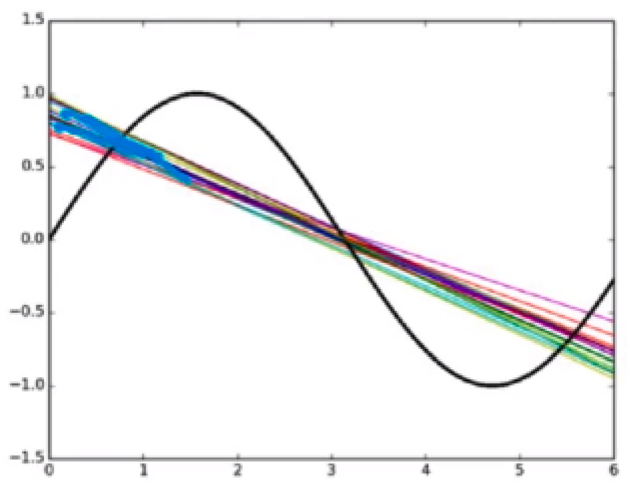


We have the input **'x'**, the true output corresponding to that input, predicted output corresponding to that input. If we average all the predicted outputs across different models(that we get by training using different subsets of training data) for a given input that would be the expected value of the model for that input. **Let's call this expected value as 'a'.** The predicted output across different models for the same input is just **'a1'**, **'a2'**, **'a3',…..** all the way up to **'an'**, then **we have the variance as average of the summation of the squared difference between the predicted value of each of the model with the expected/value across all of the models.**



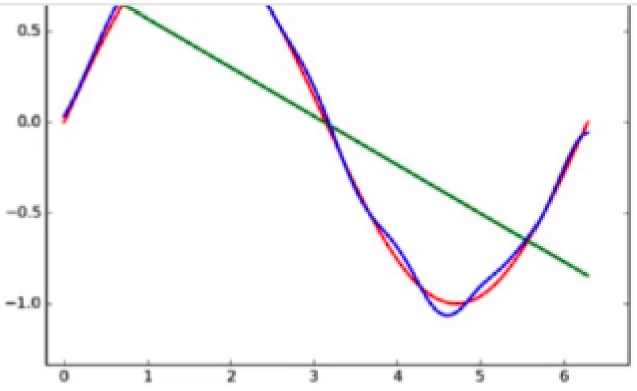
This will be for all the models that we have tried

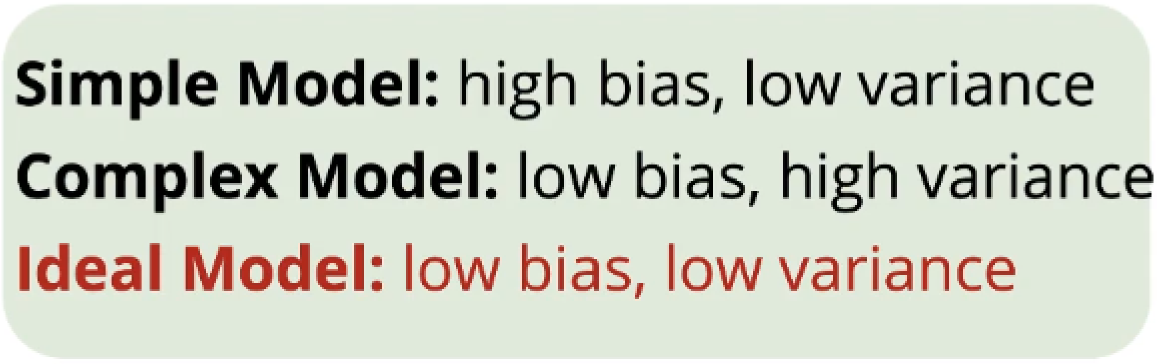
For a simple model, as all the lines are going to be very close to each other that means the average/expected value across all the models would be close to the predicted output for any of the models. That means the difference between the predicted output value and the expected value would be small and hence **we can say that the variance of the simple model is very small**. And if we look at the literal meaning, the variance tells what the spread is, we can see that the spread of a simple model is very small, they are very close to each other and in a narrow band. They all the predicting very similar values, hence they are very similar to the average, hence the distance from the average is going to be very less, and hence the variance is going to be very less.



The same thing if we do with a complex model, all the predicted outputs are going to very different from each other, as each of the curves is going to be very different, they are going to predict very different values based on the parameters that they have learned. Now any single prediction is going to be far off from the average predicted value across all the models and that's exactly what variance captures. So, **that means for a complex model, the variance is going to be high.**

Intuitively also, variance means the spread and if we look at the spread, all these curves are spread out and they are very different from each other, they are going to be very far from the average curve that we get which is the blue curve in the below plot:



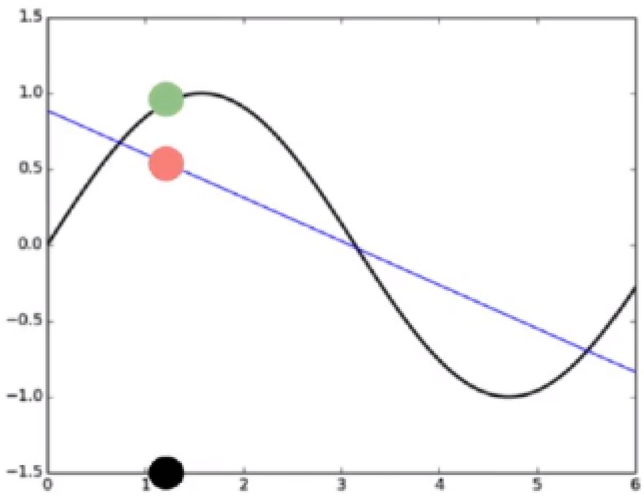

Simple Model: high bias, low variance

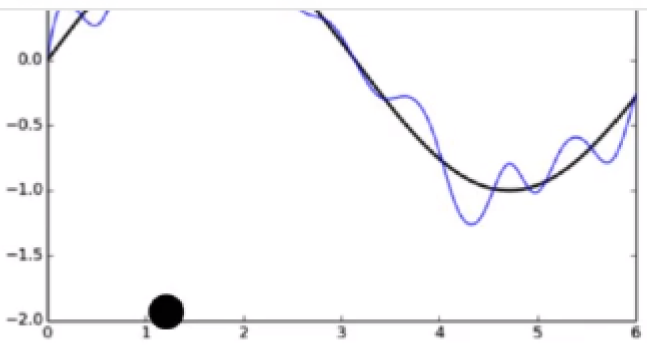
Complex Model: low bias, high variance

Ideal Model: low bias, low variance

Test Error due to high variance and high bias:

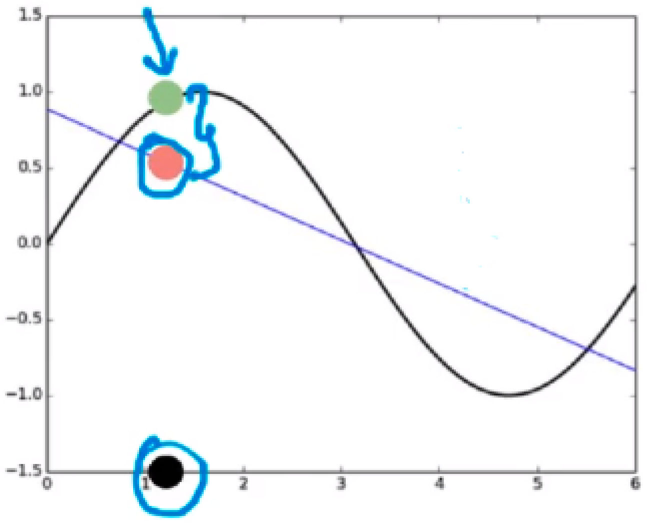
We have trained the models based on the given data and the simple model has failed miserably as the predicted output is far from the true output. On the other hand, a complex has done a good job, it was able to fit all the training points perfectly and while doing so it had to deviate from the sine function but that is acceptable as the loss was very small in that case. So, this was done on the training data.





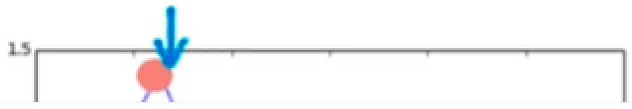
What we care about is the test error.

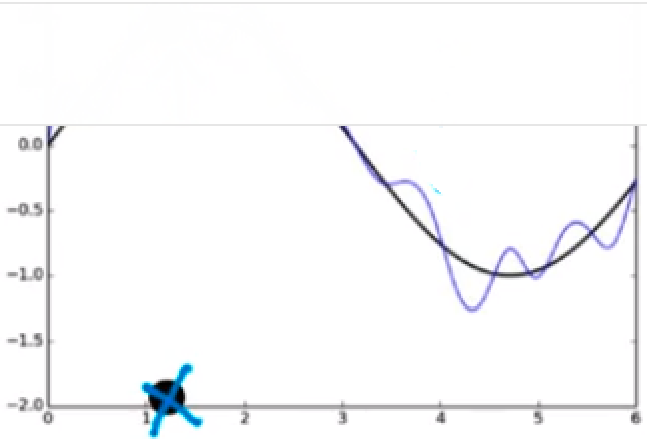
In the case of a simple model, as depicted below, if the black circle corresponds to the test set input point, the green circle corresponds to the true output and the pink circle corresponds to the output given by the simple model. Now, in this case, test error would be high given by the difference between the values represented by green and pink circles.



So, **clearly a high bias model leads to test error also**. **As the training error itself is high, it is reasonable that the test error is also going to be high because if the model could not learn from training data then it is bound to do miserably on the test data.**

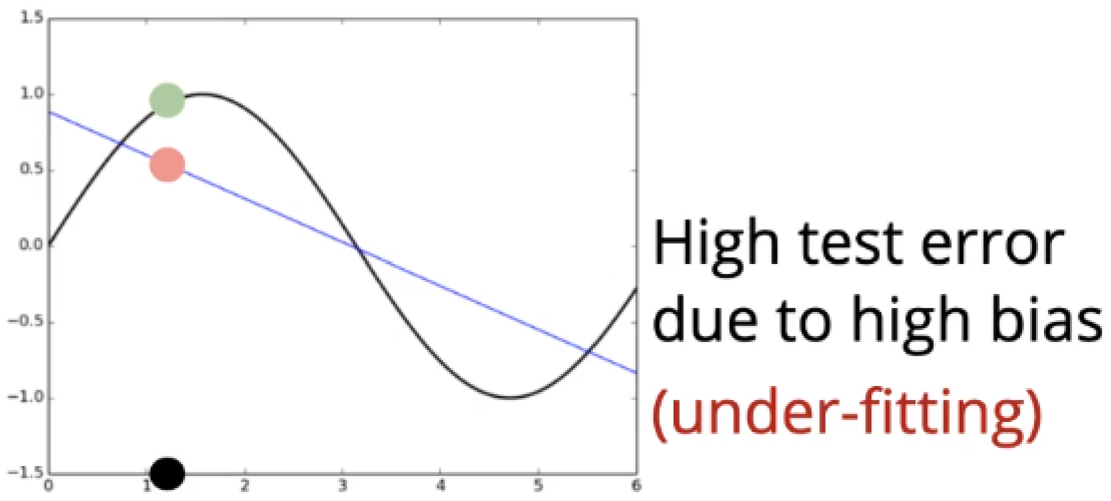
Let's see what is the situation with a complex model that has a high variance.

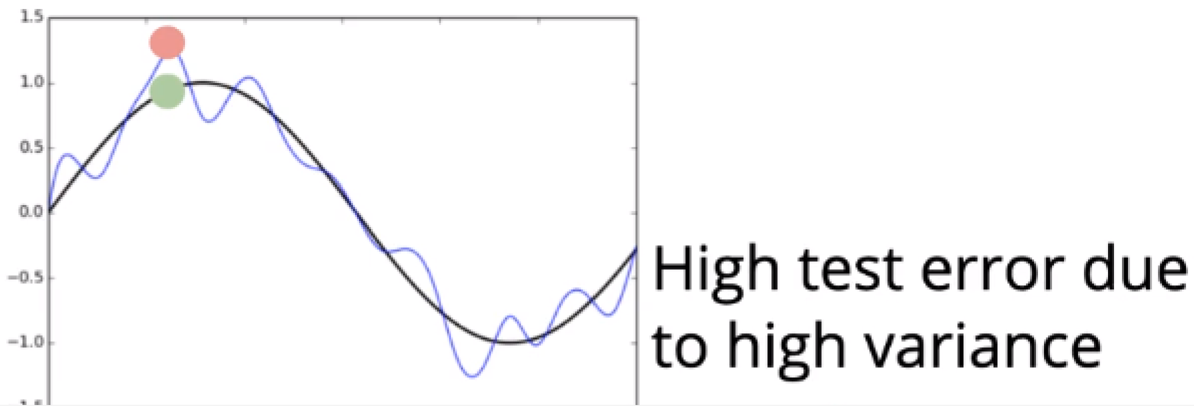


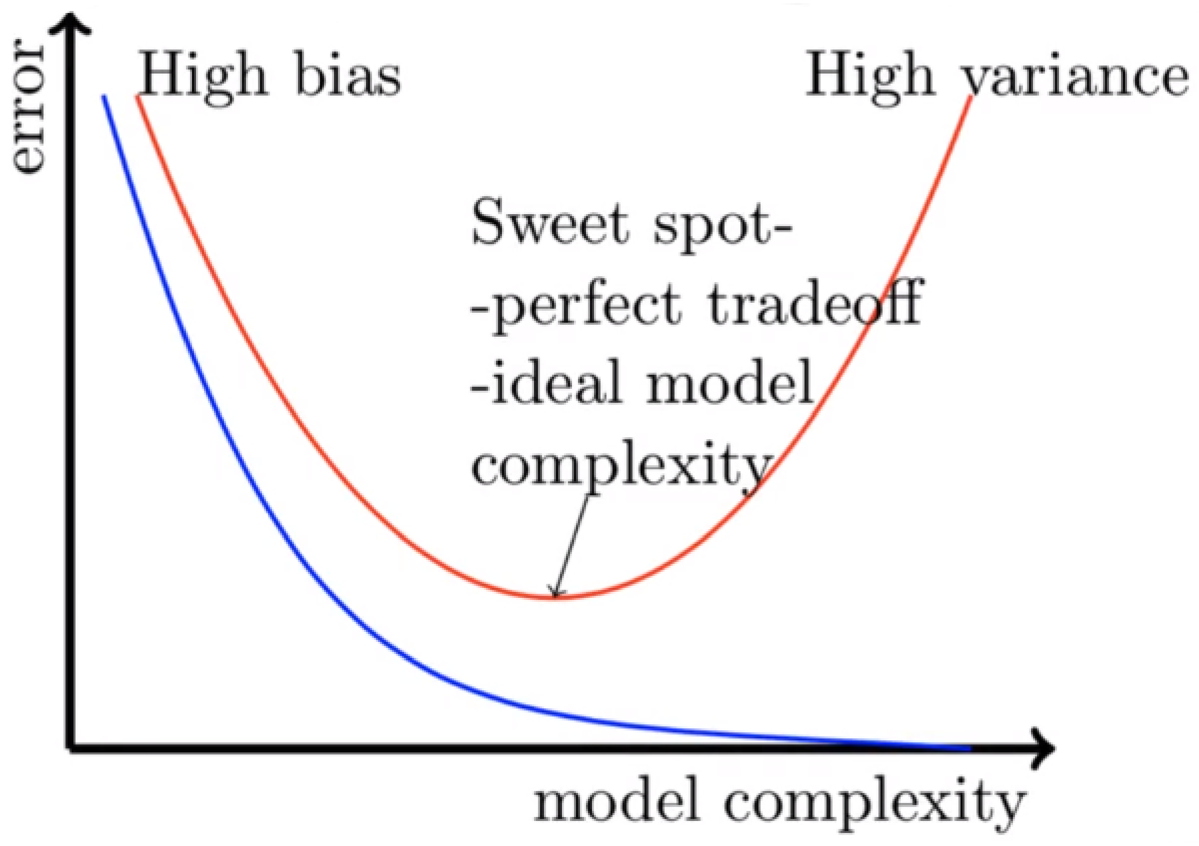


So, as is clear from the above image, **we are getting the test error on a complex model as well**. This is happening because this model focussed too much on learning the training data, it did not think what would be the case when it gets a newer training point which it has never seen. So, a high variance model is going to give a high test error.

The case with the simple model is called as under-fitting as the model could not even fit the training data whereas the case with high variance model is termed as over-fitting as the model focussed too much on fitting the training data and it did not care about how it's going to generalize for other points outside the training data.





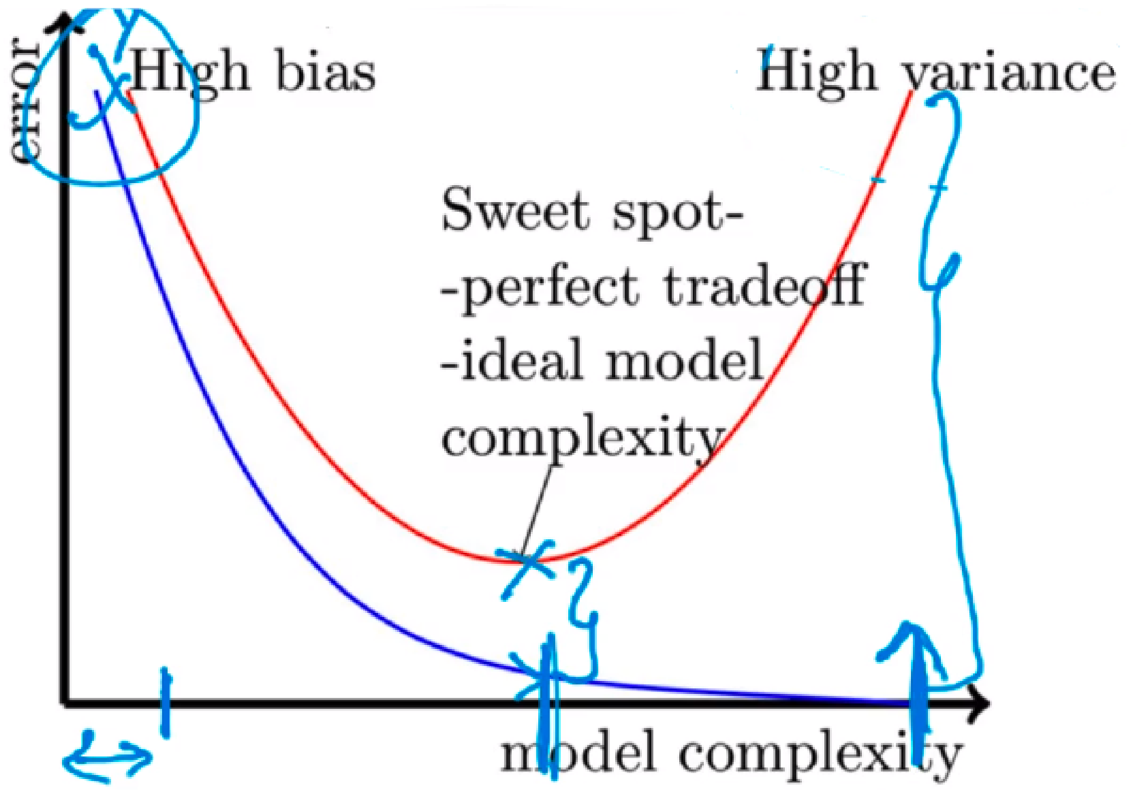


The above image shows how the training error and test error looks as we increase the model's complexity. Blueline in the above image refers to the training error and the red line corresponds to the test error.

As we increase the model complexity, the training error drops but the test error would still be high for a complex model.

When the model complexity is low(simple model), training, as well as test error, is high as discussed above in this article(case of simple model where we have a line as the model's equation) whereas on the other hand when the model is complex, training error would be very small as the complex model would fit the training data very well but the test error would be high as discussed above in this article.

Ideally, we want the sweet spot where the model's complexity is just about right so that it's doing well on the training error and the gap between the training error and the test error is not very high. So, that's the tradeoff we try to strike in ML/DL that we have the right model complexity, so that neither the bias is very high nor the variance is very high as both leads to a high test error and that's the situation we try to avoid.



Deep Learning Bias Variance Tradeoff Machine Learning Artificial Intelligence

Artificial Neural Network